Universities Space Research Association

USRA Collaborative HPC Research with NASA AMES

P. Aaron Lott

Sr. Interdisciplinary Scientist

ESTABLISHED 1969

Advancing Space Research and Exploration for 50 Years

Outline

- University-Led High Performance Computing (ULHPC) Research Program
  - A Flexible Encoding Framework and Autonomic Runtime System for Progressive Streaming of Scientific Data
  - HPC for Flight Control Planning of Multiple UAV operations for widespread applications

- HPC internship program
  - Exploring HPC I/O Capabilities for Machine Learning Use-Cases
  - Containers for Physics-based and AI/ML workloads

- AI/ML HPC services via PRP
  - Platform overview
  - Geospatial data visualization framework

USRA HPC Research Program Objectives

Program Objective

- Identify, develop, and demonstrate advanced supercomputing capabilities that would have practical value for enhancing NAS Services in a 2-5 year timeframe.
- Increase diversity, equity and inclusion and build capacity for HPC Research

University-Led High Performance Computing (ULHPC) Research Program

HPC Research Topics include but are not limited to:

- Frameworks and Compute Services
  - Frameworks for hybrid cloud/on-prem scientific workflows
  - Science data processing pipeline frameworks
  - Hardware enhancement for pathfinding systems
  - Performance prediction models
  - Benchmarks for new architectures

- HPC for Artificial Intelligence
  - Mixed/variable precision computing and machine learning/AI for scientific computation
  - AI/ML algorithms to exploit traditional hardware architectures
  - Optimized techniques/tools for visualization, data analytics, artificial intelligence and machine learning

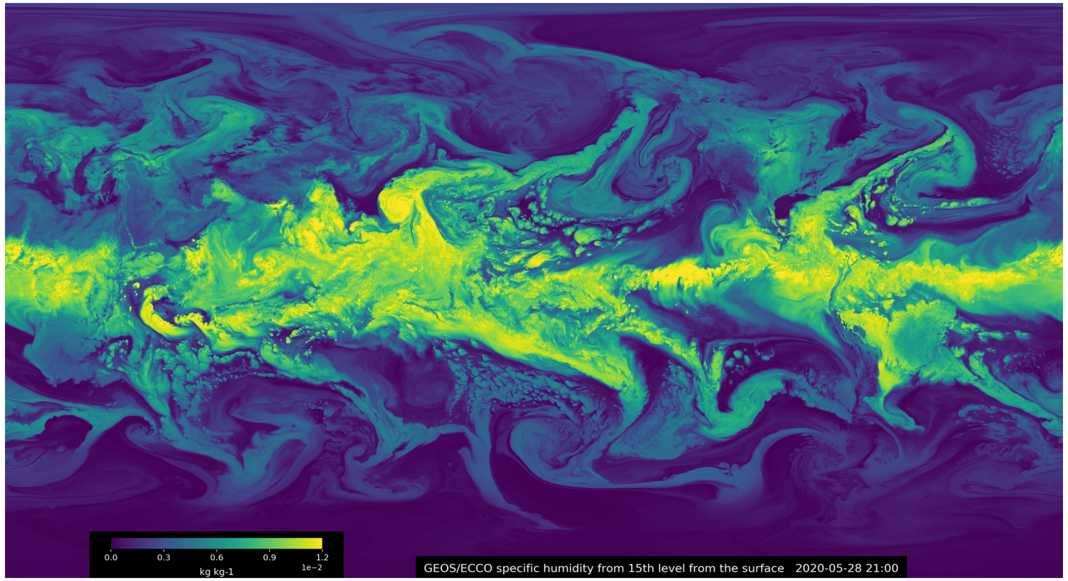
- Data Storage and Retrieval Services
  - Frameworks for distributed data depositories
  - Lossy compression for scientific datasets
  - Hardware infrastructure, file system technologies and/or application enhancements for optimized I/O

- Software and Applications
  - Programming approaches for accelerated and non-accelerated architectures
  - Efficient math libraries
  - Software packaging technology, such as containers
  - Algorithms to exploit new hardware enhancements
  - Methodologies and tools for application performance optimization

Use-Inspired HPC Research for the NAS User Community

Use-inspired HPC research - developing the future HPC workforce with multidisciplinary, multi-stakeholder teams focused on addressing some of society's grand challenges.

A Flexible Encoding Framework and Autonomic Runtime System for Progressive Streaming of Scientific Data
PI: Valerio Pascucci Utah



HPC for Flight Control Planning of Multiple UAV operations for widespread applications
PI: Subodh Bhandari, Cal Poly Pomona

A Flexible Encoding Framework and Autonomic Runtime System for Progressive Streaming of Scientific Data

Cooperative Agreement Award: 80NSSC21M0353; PI Pascucci; 21-NUP2021-0072

Credit: Valerio Pascucci

Objectives

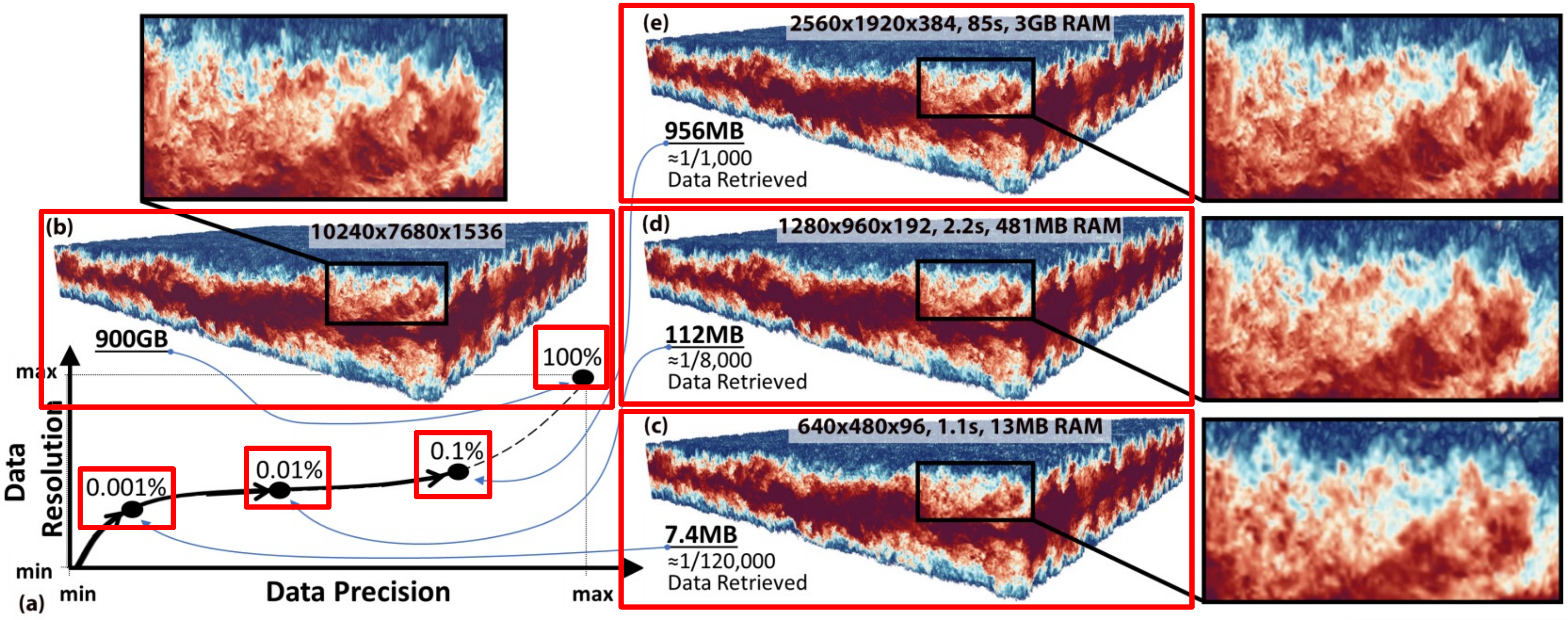
- Develop a novel data model for efficient storage and progressive streaming of large data for analysis and visualization.
- Develop a novel runtime system that automates data reduction and resource allocation decisions for in-situ workflows

Approach

- Novel representation that allows encoding scientific data at multiple levels of ***spatial resolution and numerical precision***
- From a single (lossless) encoding of the data, an application can extract multiple data streams that satisfy different tradeoffs in terms of data quality and use of the available resources available (computing, memory, network, …)

Example

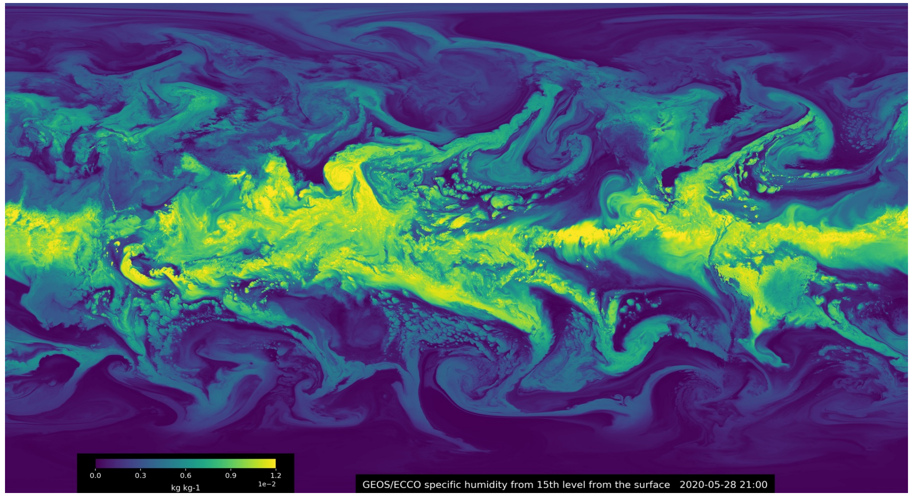
- The data resolution-precisions diagram shows how a ~1TB data stream can generate effective approximations with only 0.001% of the data while more data can always be streamed to improve the quality requirements
- **Key Technologies:** OpenViSUS, Data Spaces, National Science Data Fabric



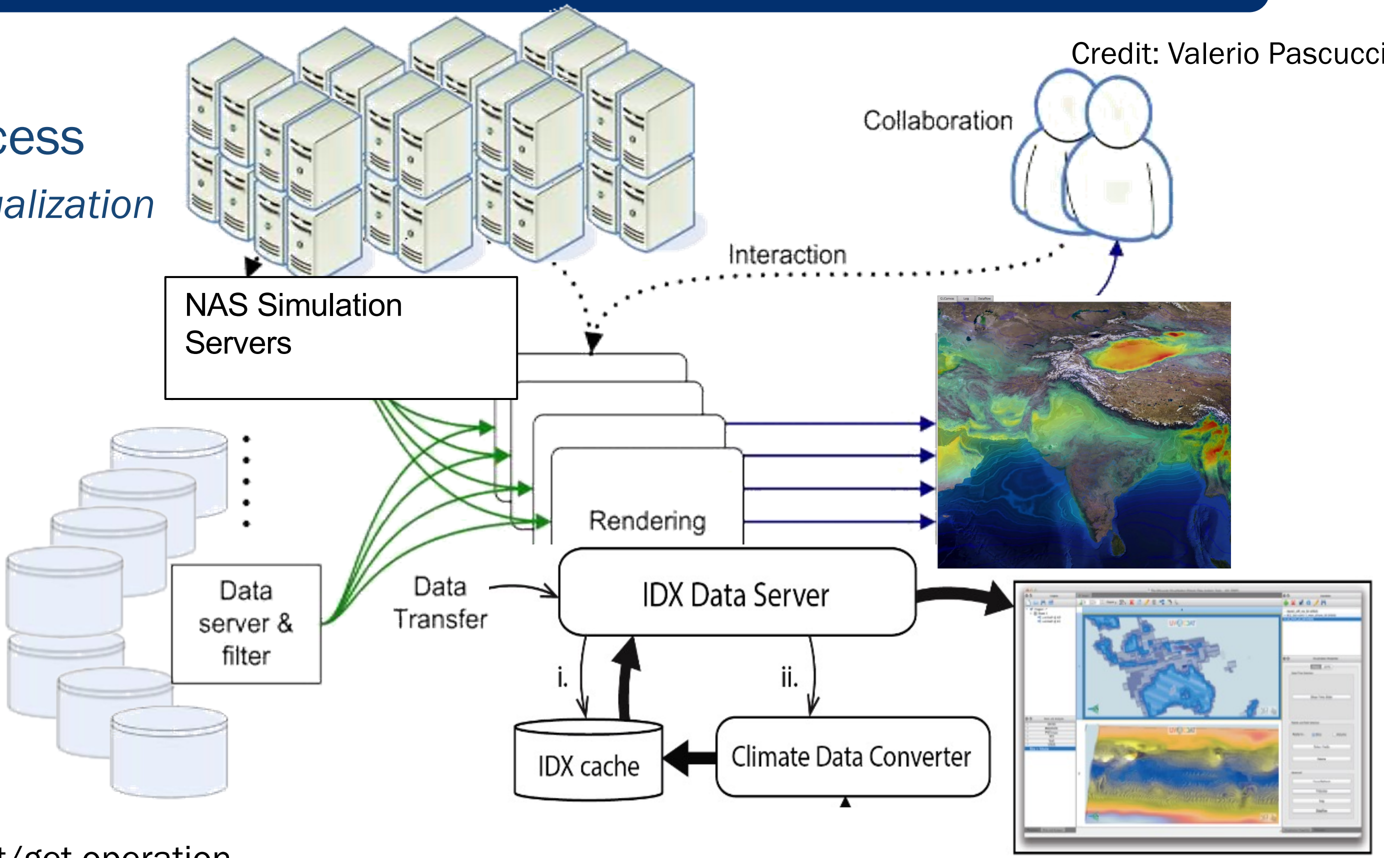
Example of Scalable Deployment: Real-time exploration of the DYAMOND coupled Ocean-Atmosphere simulation output, & ECCO's 1/48° MITgcm Ocean simulation output

Rethinking Extreme-Scale Workflows:
In-situ workflows, In-transit processing, remote access

- *Data Creation, Management, Processing, Analysis, and Visualization*



Credit: Valerio Pascucci



Autonomic workflow management system

- Support scalable in-memory data management such as the data put/get operation
- Execute customized functions (e.g., data reduction) asynchronously along with the data transfer at the place where the data is located (source, destination, or in-transit);
- Flexible configurations and organization of execution flows that contain multiple tasks in the data-staging area using publish/subscribe events
- Analyze system and workflow characteristics from previous runs to drive execution flows for workflow components, such as compression tasks

HPC for Flight Control Planning of Multiple UAV Operations

PI: Dr. Subodh Bhandari Grant No. 80NSSC21M0354

Credit: Subodh Bhandari

Goals:

- Development of capabilities for the operation/command/control, trajectory scheduling, and trajectory visualization of a large number of UAVs
- Integration of such capabilities into the National Airspace System for widespread adoptions of UAVs for civilian & commercial applications

UAVs are being targeted for many dull, dirty, and dangerous applications

- Fire detection, monitoring, and suppression
- Search and rescue/surveillance of disaster-hit areas
- Traffic monitoring
- Precision agriculture/crop dusting

UAVs have limited payload capacities and endurance
Realistic mission scenarios require UAVs need to be operated simultaneously

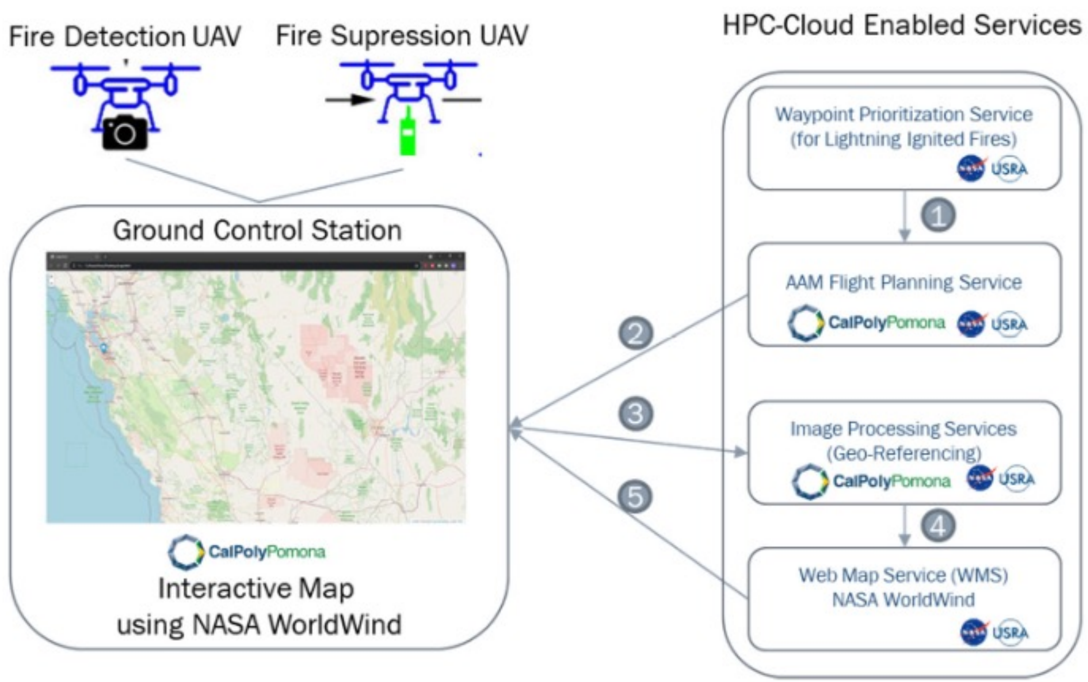
Scalable operation, requires

- Command and control multiple UAVs from a single ground control station
- Autonomous coordination between UAVs for payload distribution

HPC is needed for simulation & operation of a large number of UAVs

- Increased level of autonomy
- Trajectory scheduling & optimal path planning
- Autonomous collision detection and avoidance
- Real-time situational awareness for ground crews in hazardous areas

e.g. accurate 3-D maps



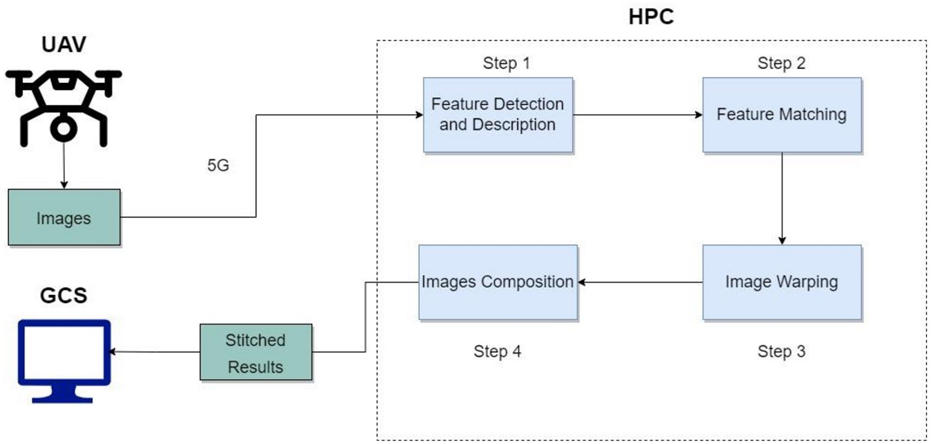
HPC for Flight Control Planning of Multiple UAV Operations

Credit: Subodh Bhandari

UAVs work in teams to collect data from multiple channels

Information is transmitted to HPC system running multiple algorithms for analysis

Channel data must be stitched together for analysis and stitched results transmitted back to control center for updated UAV planning/scheduling



HPC for Flight Control Planning of Multiple UAV Operations

Credit: Subodh Bhandari

- Current Developments:
  - Development of ground control station for command and control of multiple UAVs
  - Interactive map using NASA WorldWind
  - Multi-UAV trajectory generation/path planning methods
  - Coordination between multiple vehicles
  - Improvement in fire detection algorithm/image stabilization
  - Multi-UAV trajectory optimization
  - Ortho-mosaicking of RGB images of the scanned area/thermal image overlay
  - 3-D forest fuel map generation using LiDAR to predict fire behavior

HPC for Flight Control Planning of Multiple UAV Operations "Stack"

Resource Requirements:

- **OS**: 64 bit.
- **Programming Language**: Python, C++
- **Environment**: Python Integrated Development Environment.
- **Memory:** 10G
- **Storage:** 1-2 T
- **I/O**: both static and dynamic
- **Math Libraries**: FFT, MKL
- **Computer Vision/Image Processing Libraries:** OpeCV, Pillow
- **Any External Libraries for data**: TBD
- **Parallel Processing**: MPI, OpenMP, NVDIA CUDA-X
- **Compilers:** nvcc, g++
- **MPI**: HPE's MPT
- **Data Analysis Tools:** TBD
- **Machine Learning libraries or tools**: PyTorch, Scikit-Learn, TensorFlow

HPC Tasks:

- Real- or near real-time processing of UAV data (thermal, RGB, and LiDAR)
- Investigate the effect of large volume of data processing and simulation on I/O bottlenecks in HPC and stability of GPUs.

Current HPC Framework:

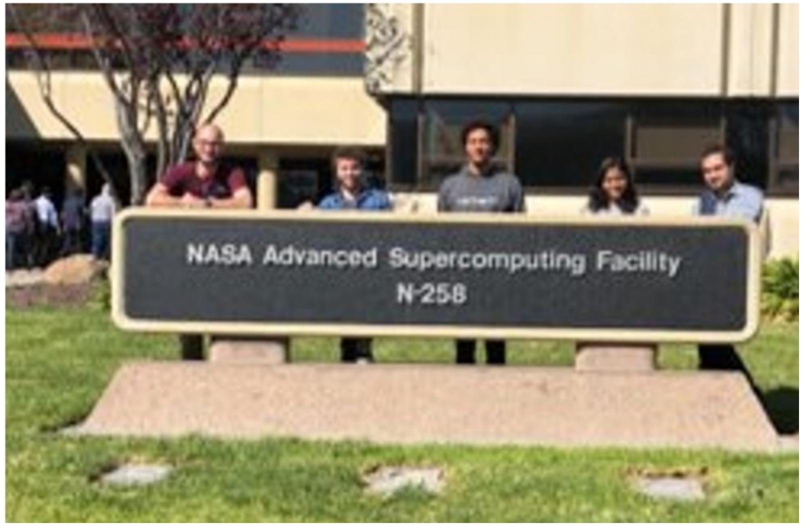
- PRP integration

Assessing Bottlenecks for Real-time NAS -UAV Wildfire Service:

- Possibilities of real-time connection between UAVs and NAS HPCs using VPN
- Software installation/availability allowed on NAS

HPC Internship program

- We are seeking student interns to conduct collaborative research and workforce development projects related to high performance computing. Internship projects directly contribute to USRA and NASA’s mission, with diverse STEM projects.
- The involvement in this research would not only allow for the hands-on application of learned skills, but would also improve the communication of research findings in the form of code, scientific publications and/or presentations.
- Multiple High-Performance Computing (HPC) internships are available. The objective of the internships will be to identify, develop, and (where appropriate) demonstrate advanced supercomputing capabilities that would have practical value for enhancing NAS Services in a two- to five-year timeframe. NAS Services include high-end computing, storage, networking, and associated capabilities that enable scientists and engineers supporting NASA missions in space exploration, scientific discovery, and aeronautics research.

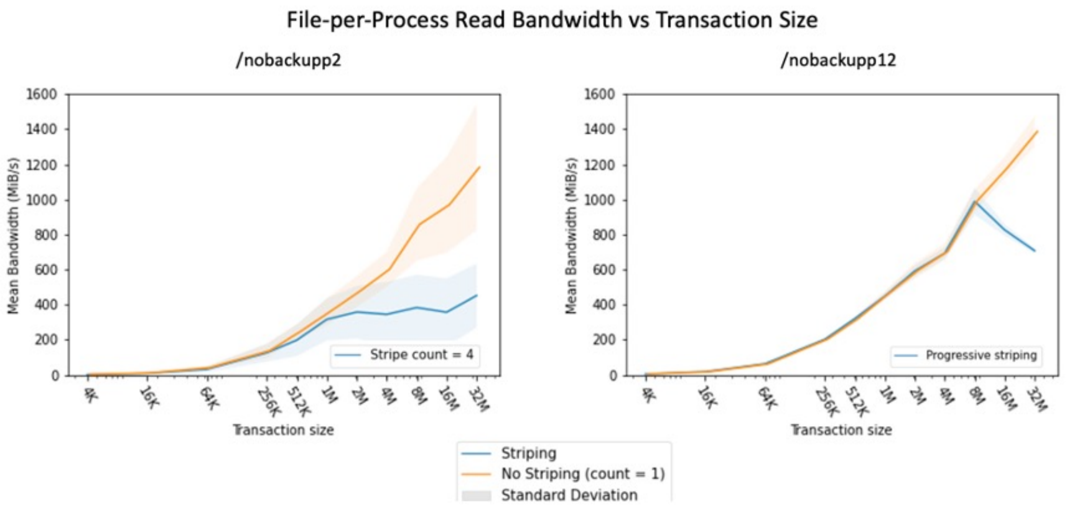


DESIRED QUALIFICATIONS

- Enrolled in a degree program in Computer Science, Data Science/Machine Learning, Mathematics, Computational Science,Artificial Intelligence or related field
- Experience with Python, Docker, Kubermetes, Jupyter, Artificial Intelligence, Machine Learning, Deep learning, and analyzing performance in High Performance Computing (HPC) environments

Collaborative Student Research Projects

Exploring HPC I/O Capabilities for Machine Learning Use-Cases
Parker Addison (USCD), David Bell (USRA), Aaron Lott (USRA), Henry Jin (NASA Ames)



Containers for HPC & Machine Learning workflows
Caetano Malone (Stanford), David Bell (USRA), Aaron Lott (USRA), Henry Jin (NASA Ames)

Exploring HPC I/O Capabilities for Machine Learning Use-Cases

Parker Addison (USCD), David Bell (USRA), Aaron Lott (USRA), Henry Jin (NASA Ames)

- Adoption of data-driven machine learning applications on high-performance computing (HPC) systems has grown considerably.

- Machine learning applications demand different data access (I/O) behavior than simulation-based applications which have historically informed HPC system design.

- We evaluate the I/O performance of data-driven machine learning use-cases in established scientific HPC environments.

- We evaluate data access patterns in 3 ML applications then profile workloads via IOR
  - Preprocessing data & Training a VAE for flight anomaly detection via black box data
  - Preprocessing wildfire perimeters & Training a linear regression model for wildfire risk estimation based on satellite and lightning strike data
  - Training a deep CNN for flood detection based on satellite remote sensing data & performing inference on remote sensing data

We profile the ML workloads with Darshan. Darshan is a comprehensive application I/O profiling tool developed by Argonne National Laboratory for use on HPC systems.

Use IOR to characterize I/O performance
Operation timings and access bandwidths

Exploring HPC I/O Capabilities for Machine Learning Use-Cases

During preprocessing the full dataset must be read which can lead to I/O bottlenecks,

Some workflows in the training stage exhibit similar I/O demands, but there is a notable difference between statistical learning and deep learning applications.

Statistical learning such as that in the *wildfire* use-case relies on a closed-form mathematical expression to produce a trained model and thus **only needs to read the training dataset once.**

Deep learning such as that in the *flight* and *flood* use-cases relies on an **iterative** approach to solve for an optimal model and thus require **multiple ingestions of the training dataset.**

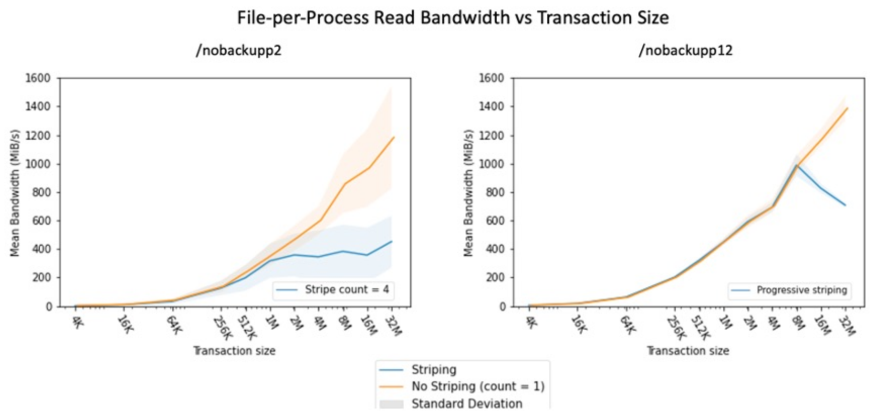
Even within the realm of deep learning training, differences in the **volume** of the training dataset produces different I/O performance. Sufficiently **small datasets can be stored in the page cache and benefit from incredibly quick re-reads**, whereas **datasets larger than the available memory cannot be fully cached** and suffer an I/O performance impact.

Inference requires preprocessing and model loading for a given set of input data, in which case we determined the **velocity at which new input data arrives** (and at which the result of inference is required) determines whether current I/O performance poses an issue.

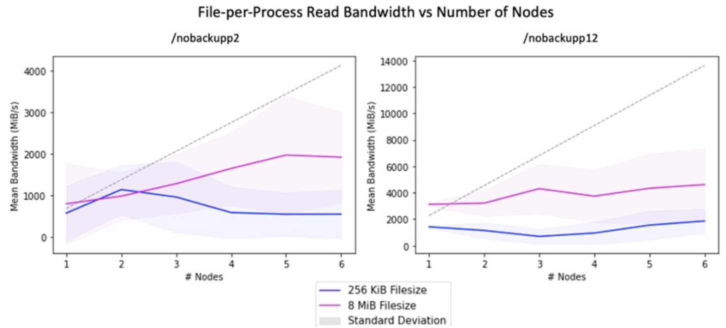
Use case	Workflow			I/O sizes	Data Volume	Read Count	Performance Notes
	Prep	Train	Inference				
flight	X			Small	>=Mem	Once	I/O bottleneck
flight		X		Large	<=Mem	Many	Dataset cached
wildfire	X			Small	>=Mem	Once	Compute dependent
wildfire		X		Small	<=Mem	Once	Statistical learning
flood		X		Small	Varies	Many	Volume dependent
flood			X	Small	Varies	Once	Velocity dependent

Exploring HPC I/O Capabilities for Machine Learning Use-Cases

- Data preprocessing for machine learning use-cases that demand a **single full-dataset read** will exhibit **an I/O bottleneck.**
- Deep Learning training that demand **multiple full-dataset reads** may exhibit an **I/O bottleneck if the data volume is greater than the available memory**.
- Inference on machine learning models may exhibit an **I/O bottleneck if a large volume of data** is required with **sufficiently high velocity.**
- **Lustre striping** is designed to optimize **multiple-processor access to shared files** and to support storage of large file sizes. Though classical HPC applications may make use of this practice, it adds networking overhead and server contention for machine learning representative workflows of moderately small single-processor file access.
- Fine-grained I/O demand characterization of machine learning workflows is a developing field. Although **Darshan** is proven on classical HPC applications, **current incompatibilities** with **Python multiprocessing and TensorFlow data** APIs pose a significant barrier to effective characterization of machine learning applications.
- IOR is useful for gaining a general understanding of filesystem characteristics, but the tool cannot fully replicate the many-file access and metadata demands of machine learning workflows.



Single processor read bandwidth with and without Lustre striping, as file and I/O transaction size increases. IOR benchmarks testing filesystem performance at various transaction sizes confirm that read bandwidth suffers at small file sizes for all file formats and filesystem configuration. POSIX file-per-processor access to moderate sized files (ones to tens of megabytes) is negatively affected by Lustre striping.



Read bandwidth at small and moderate transaction sizes as number of participating nodes increases.

Containers for HPC & ML Workflows

Caetano Melone, Stanford University

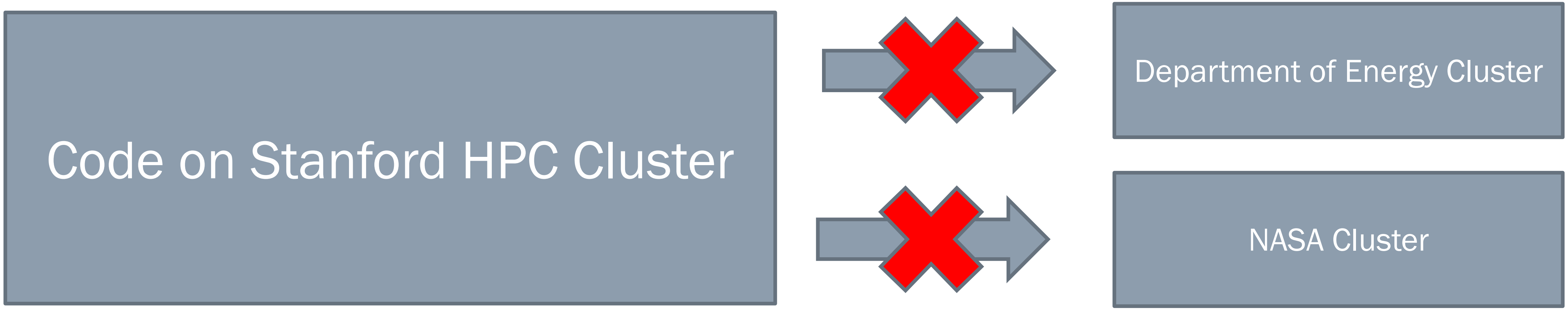
- Containers are virtualized operating systems, allowing users to build custom environments that can be easily transferred for execution on heterogeneous High-Performance Computing (HPC) systems

Objectives

- Investigate how containers are being used on HPC platforms at organizations such as the Department of Energy
- Leverage tools released by the Extreme-scale Scientific Software Stack (E4S) project
- Apply best practices and new technologies on NAS machines, verifying their performance and writing documentation for users

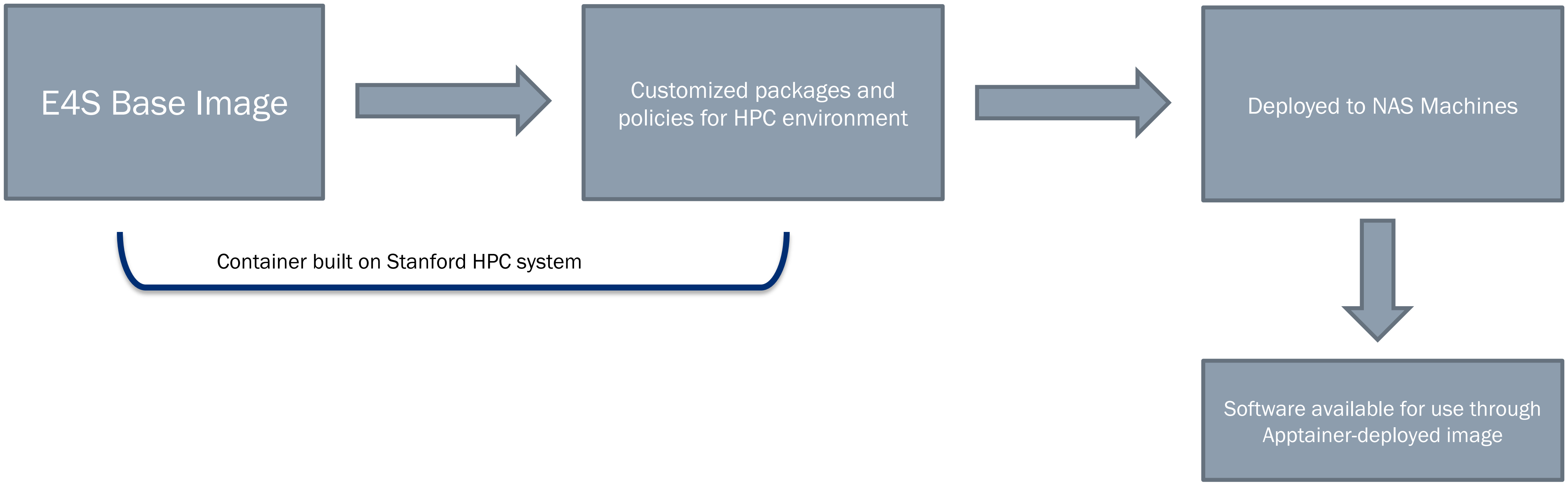
Portability

- Software is distributed through **modules**, offering ease of use but little flexibility
  - System administrators must update these packages themselves
  - Users also have the option to manually compile software, adding complexity
- Portability is often an issue



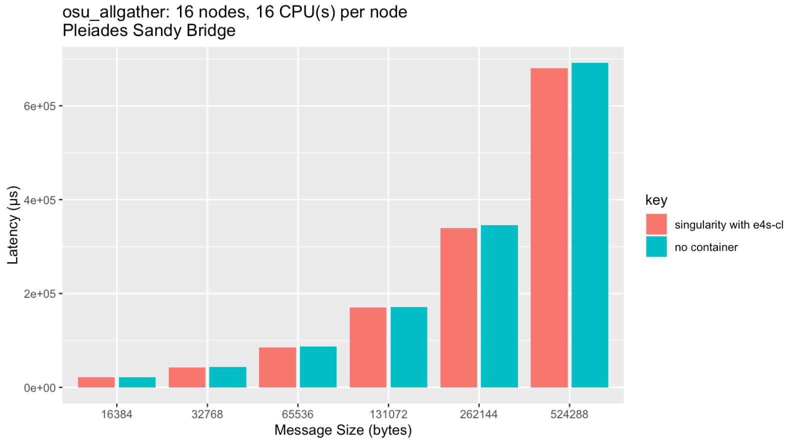
Containers for HPC Workflows

- Based off the DOE Extreme-scale Scientific Software Stack (E4S) project, built custom image and deployed on NAS systems
  - Image optimized for standard HPC policies and other user considerations



Recent Achievements

- Investigated the running of multi-node jobs
  - Running Message Passing Interface (MPI) apps with Singularity and Podman requires extra steps
  - **Important for the encouragement of adoption among NAS users**
  - Using tool developed by E4S team, have verified a solution to this problem, allowing these programs to be run with ease
- Performance testing of CPU and GPU jobs on containers
  - Containers on par with bare metal workflows
- Building a series of containers to support ML workflows
  - Supporting every step of the process: data processing, training, etc.



Machine Learning Containers
Earth Science Focus

- Each built for a specific purpose, containing software to perform these tasks
- E4S base images as a foundation (lightweight)

1. Data Pipeline
   - GDAL, xarray, PyTorch
2. Modeling
   - PyTorch, TensorFlow
3. Evaluation
   - MLFlow, TensorBoard
4. Model Engineering
   - PyTorch, MLflow, skorch
5. Model Packaging
   - MLflow
6. Serving

AI/ML HPC services via PRP

Digital Earth Geospatial Visualization Service

- USRA's Research Institute for Advanced Computer Science (RIACS) provides access to high-performance GPU and storage nodes for select university research projects aligned with USRA activities. This includes USRA-owned compute and storage resources, as well an extended portfolio of CPU and GPU nodes made available through a partnership with Pacific Research Platform (PRP).

- PRP is a partnership between more than 50 institutions including research universities. Led by researchers at UC San Diego and UC Berkeley, PRP is creating a seamless research platform that encourages collaboration on a broad range of data-intensive fields and projects.

PRP Nautilus HyperCluster Architecture

- PRP's Nautilus is a HyperCluster for running containerized Big Data Applications. It utilizes Kubernetes for managing and scaling containerized applications in conjunction with Rook for automating Ceph data services.

kubernetes ROOK

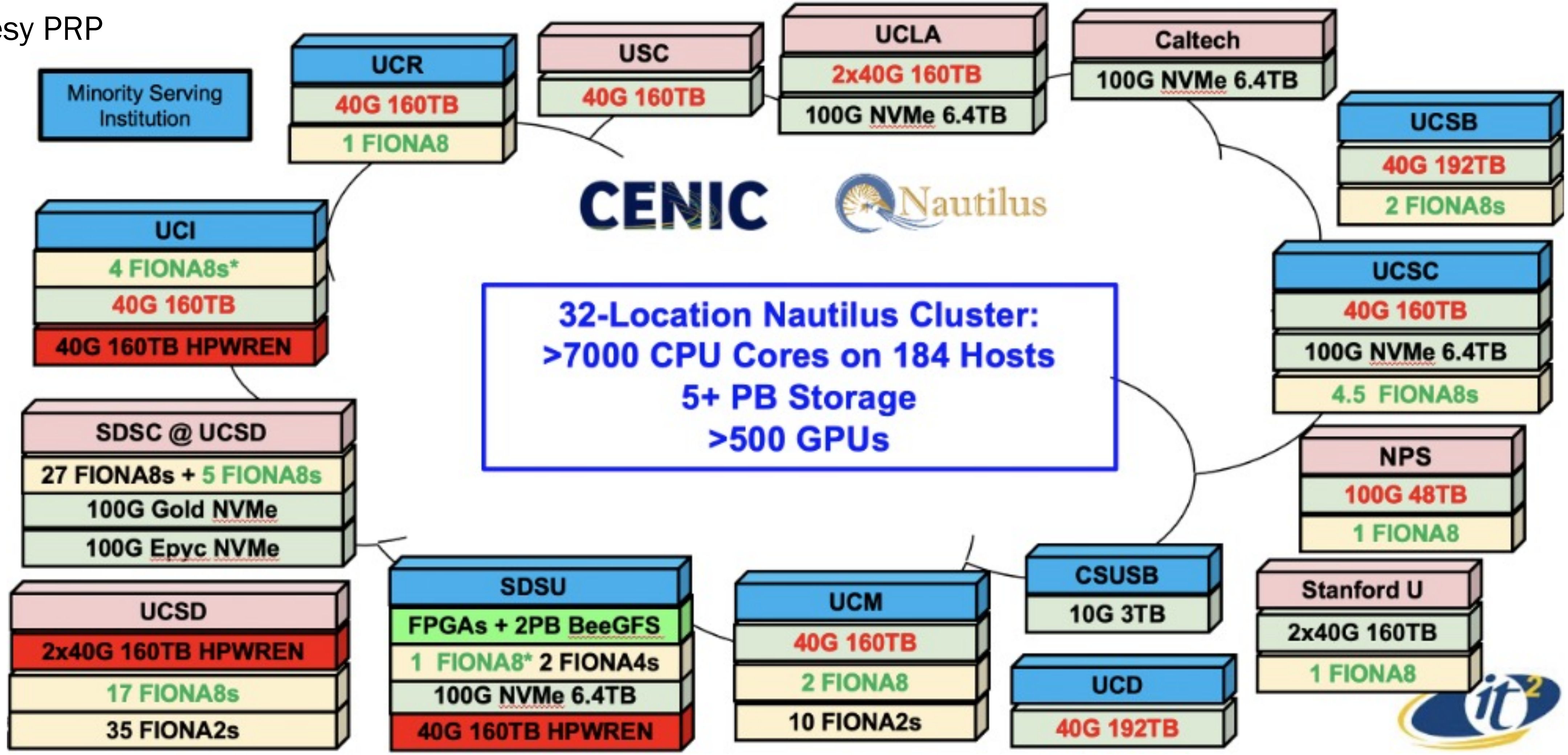
- Kubernetes, is an open-source system for automating deployment, scaling, and management of containerized applications.

- Rook is an open-source file, block, and object storage for cloud-native environments. For PRP, Rook runs inside Kubernetes to manage petabytes of distributed storage and CPUs for data science while PRP measure and monitors network use.

Courtesy PRP

PRP Nautilus HyperCluster Resources (West Coast)

Courtesy PRP



PRP Data Transfer Nodes (DTNs): Flash I/O Network Appliances (FIONAs)

- Flash Input/Output Network Appliance (FIONAs) are rack-mounted PCs designed by the PRP team. They are Science DMZ Data Transfer Nodes (DTNs), optimized for **10-100Gbps data transfers** and large data storage, and are also **capable of holding up to eight GPUs or FPGA add-in boards or up to 256TB of disk each.**

Two FIONA DTNs at UC Santa Cruz: 40G & 100G
Up to 192 TB Rotating Storage

Add Up to 8 Nvidia GPUs Per 2U FIONA
To Add Machine Learning Capability

- PRP has 180 FIONAs on 25 partner campuses networked together at 10-100Gbps.
- PRP UCSD-Designed FIONAs Solved Disk-to-Disk Data Transfer Problem *at Near Full Speed* on Best-Effort 10G, 40G and 100G Networks.

Courtesy PRP

PRP Field Programmable Gate Arrays (FPGAs)

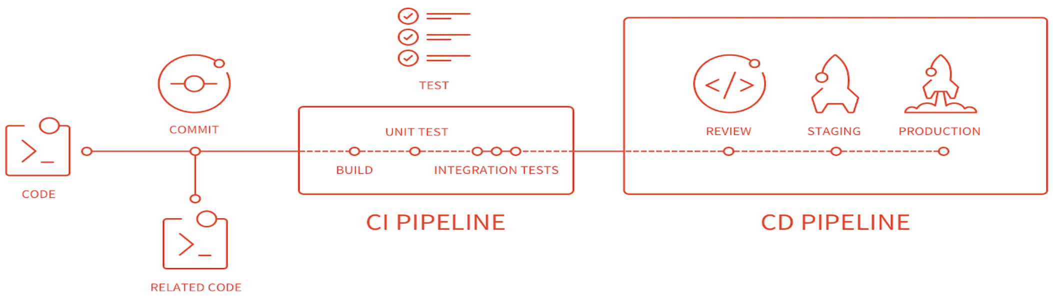
- **Field Programmable Gate Arrays** (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing by customer or designer.

Courtesy PRP

- The **Xilinx FPGA** device plugin for Kubernetes is a Daemonset deployed on the Kubernetes cluster which allows you to:
  - Run **FPGA** accessible containers in the Kubernetes cluster
  - Discover the **FPGAs** inserted in each node of the cluster and expose info of the **FPGAs** such as quantities, DSA (shell) type and timestamp, etc.

PRP Cluster Management Automation: GitLab CI/CD, Ansible, Operators

- GitLab CI/CD is a tool for software development using the continuous methodologies: **Continuous Integration (CI), Continuous Delivery (CD), Continuous Deployment (CD)**
- GitLab CI/CD can automatically build, test, deploy, and monitor your applications by using Auto DevOps.



- Ansible is an open-source software provisioning, configuration management, and application-deployment tool enabling infrastructure as code

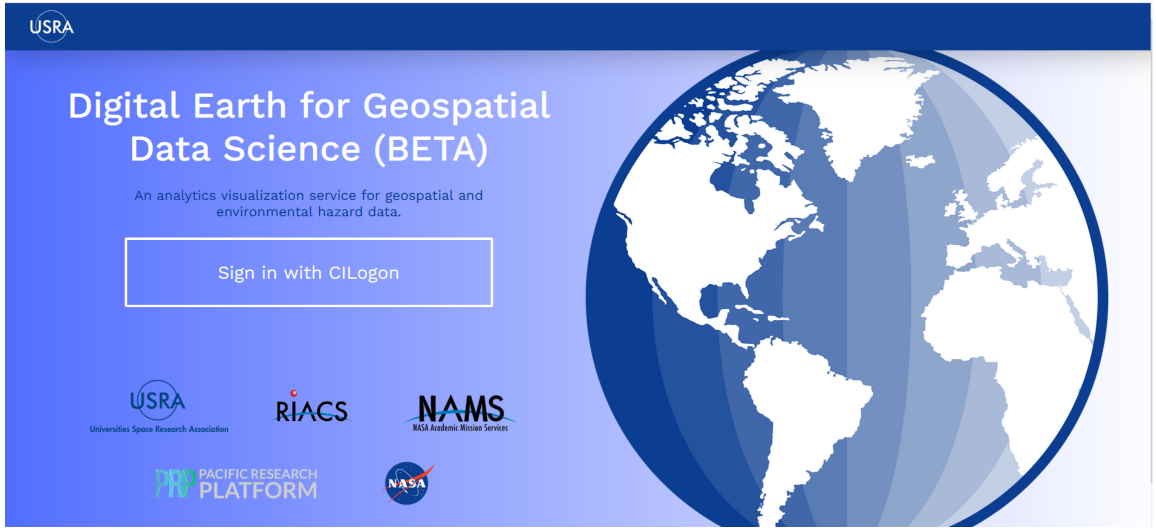
Courtesy PRP

Kubernetes Operators deployed:

- Prometheus+Thanos – Monitoring nodes and workloads
- Rook (Ceph) – Storage
- Kubevirt – VMs
- Admiralty -- Federation
- User portal
- Elasticsearch
- CertManager

Digital Earth - Geospatial Data Visualization Service

Digital Earth for Geospatial Data Science- An analytics visualization service for geospatial and environmental hazard data.



Digital Earth aims to reduce barriers for universities and collaborators to perform research in geospatial data science.

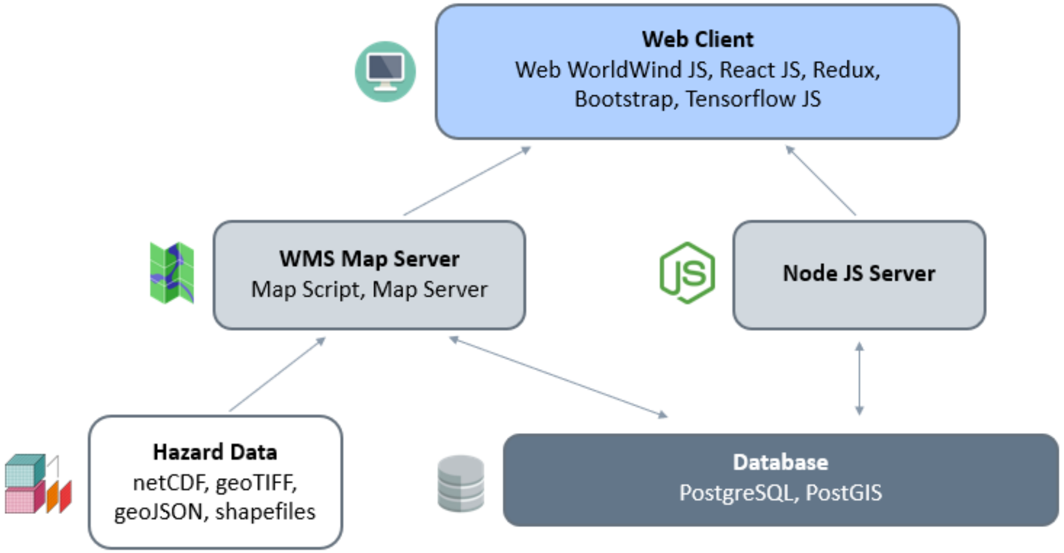
Digital Earth - Geospatial Data Visualization Service

Digital Earth is a geospatial data science visualization service based on **NASA's WorldWind Software Development Kit (SDK)** that can be utilized by various stakeholders to support client decision-making processes:

- **Pomona wildfire UAV team:** Depicting analyses of historical wildfire data to plan future actions such as developing high resolution remote sensing flight plans. Using as a staging platform for NAS experiments.
- **Civil Air Patrol (CAP) wildfire missions:** Depicting analyses of rainfall intensity maps to direct observations of field scientists for assessing post wildfire risks and supporting recovery to prevent flooding.
- **NASA Data and Reasoning Fabric (DRF) data service:** An advanced air mobility marketplace, that will utilize digital earth to visualize geospatial data services from clients

Digital Earth - Geospatial Data Visualization Service

- Foothill College, Science Learning Institute (SLI) student interns for USRA are developing the Digital Earth framework for geospatial data science using USRA HPC resources through PRP:



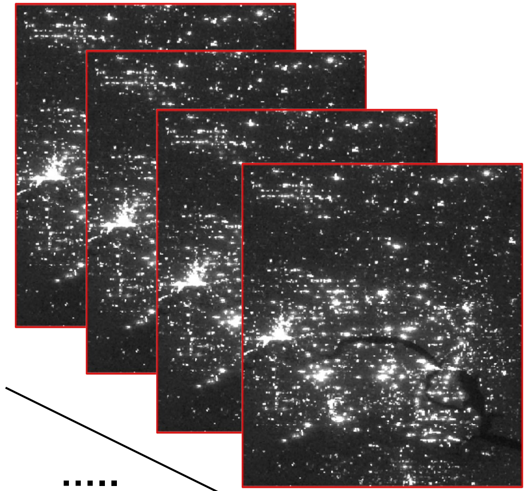
This project benefits from PRP by:

- Building and accessing mounted volumes available across the cluster housing existing or requested geospatial data by client.
- Using PRP Nautilus-provided host domains to deploy and host service, configuring porting.
- Using memory and CPU from PRP to run and maintain service.
- Using Kubernetes container images for base OS or custom-built images for software base.

Continual monitoring of the Earth at Night using NASA's Black Marble Product Suite

*Román, M.O et al. 2018. NASA's Black Marble nighttime lights product suite. *Remote Sensing of Environment*, *210*, pp.113-143.

Black Marble Product Suite*



2012

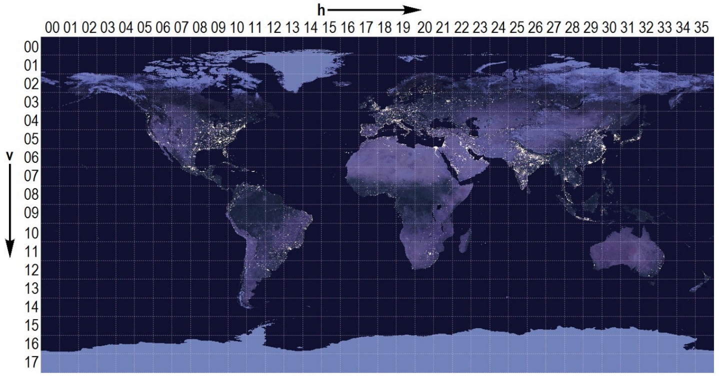
.....

present

- Daily nighttime observations in Day/ Night Band (DNB)
- Aligned VIIRS thermal bands
- Aligned masks, quality layers
- Multiple product levels for targeted science analysis
  - urbanization, disaster impact, electrification, emissions (gas flares, fires, cities), clouds, etc.

Data volume, and computational requirements for monitoring are a bottleneck

Computational Workflow



LAADS

595 daily gridded tiles at 15 arc seconds on LAADS

https://ladsweb.modaps.eosdis.nasa.gov/

Data fetching

Data transformation

Analysis

LAADS → Nautilus S3 Storage → Nautilus S3 Storage: preprocessed data → PRP Cluster: **Analysis and Inference**

stacking, masking, filtering into zarr files using docker containers for multifaceted use

containerized analysis tasks

Reducing computational and storage bottlenecks using the Nautilus environment